What Is Claimed Is:

1. A method for protecting a server against denial-of-service attacks, comprising:

receiving a request for service at the server, wherein the request is received from a client;

in response to the request, sending a random number, y, and an identifier, id_1, to the client;

allowing the client to compute a preimage, x, such that $y = h(x)$;

receiving an answer from the client, including the preimage x and an identifier, id_2;

verifying that the identifier, id_1, sent to the client matches the identifier, id_2, received from the client;

if the identifiers match, computing $h(x)$; and

if $h(x) = y$, performing the requested service for the client;

whereby the server avoids computing $h(x)$ until the server receives the answer with a matching identifier.

2. The method of claim 1, wherein the server sends a parameter, n, along with the random number y to the client, wherein the parameter n varies the amount of computational work involved in computing the preimage x.

3. The method of claim 2, wherein the parameter n specifies that a subset of n bits of $h(x)$ has to match a corresponding subset of n bits of y.

4. The method of claim 1, wherein computing the preimage, x, takes more computational effort than computing $h(x)$, whereby the client is forced to

perform more computational work than the server before the server performs the requested service.

5. The method of claim 1, wherein if $y \neq h(x)$, the server ignores subsequent communications from the client.

6. The method of claim 1, wherein if $y \neq h(x)$, the server becomes slower in responding to subsequent communications from the client, distinguished from other clients, as by its source IP address.

7. The method of claim 6, wherein each time the server determines $y \neq h(x)$, the server doubles the service time for the client, distinguished from other clients, as by its source IP address, so that the server spends progressively less time servicing requests for the client.

8. The method of claim 1,

wherein sending the random number, y, and the identifier, id_1, to the client involves first,

generating the random number y and the identifier id_1; and

storing the random number y and the identifier id_1 at the server; and

wherein verifying that id_1 matches id_2 involves first looking up id_1 and the random number y at the server.

9. The method of claim 1, wherein $h(x)$ is a hash function.

10. The method of claim 1, wherein the identifier, id_1, is inferred from data related to the communication.

11. A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for protecting a server against denial-of-service attacks, the method comprising:

receiving a request for service at the server, wherein the request is received from a client;

in response to the request, sending a random number, y, and an identifier, id_1, to the client;

allowing the client to compute a preimage, x, such that $y = h(x)$;

receiving an answer from the client, including the preimage x and an identifier, id_2;

verifying that the identifier, id_1, sent to the client matches the identifier, id_2, received from the client;

if the identifiers match, computing $h(x)$; and

if $h(x) = y$, performing the requested service for the client;

whereby the server avoids computing $h(x)$ until the server receives the answer with a matching identifier.

12. The computer-readable storage medium of claim 11, wherein the server sends a parameter, n, along with the random number y to the client, wherein the parameter n varies the amount of computational work involved in computing the preimage x.

13. The computer-readable storage medium of claim 11, wherein the parameter n specifies that a subset of n bits of $h(x)$ has to match a corresponding subset of n bits of y.

14. The computer-readable storage medium of claim 11, wherein computing the preimage, x, takes more computational effort than computing $h(x)$, whereby the client is forced to perform more computational work than the server before the server performs the requested service.

15. The computer-readable storage medium of claim 11, wherein if $y \neq h(x)$, the server ignores subsequent communications from the client.

16. The computer-readable storage medium of claim 11, wherein if $y \neq h(x)$, the server becomes slower in responding to subsequent communications from the client, distinguished from other clients, as by its source IP address.

17. The computer-readable storage medium of claim 16, wherein each time the server determines $y \neq h(x)$, the server doubles the service time for the client, distinguished from other clients, as by its source IP address, so that the server spends progressively less time servicing requests for the client.

18. The computer-readable storage medium of claim 11,

wherein sending the random number, y, and the identifier, id_1, to the client involves first,

generating the random number y and the identifier id_1; and

storing the random number y and the identifier id_1 at the server; and wherein verifying that id_1 matches id_2 involves first looking up id_1 and the random number y at the server.

19. The computer-readable storage medium of claim 11, wherein $h(x)$ is a hash function.

20. The computer-readable storage medium of claim 11, wherein the identifier, id_1, is inferred from data related to the communication.

21. An apparatus that protects a server against denial-of-service attacks, comprising:

the server;

a receiving mechanism within the server that is configured to receive a request for service from a client;

an access mechanism, wherein in response to the request, the access mechanism is configured to,

send a random number, y, and an identifier, id_1, to the client,

allow the client to compute a preimage, x, such that $y = h(x)$,

receive an answer from the client, including the preimage x and an identifier, id_2, and to

verify that the identifier, id_1, sent to the client matches the identifier, id_2, received from the client,

wherein if the identifiers match, the access mechanism is configured to compute $h(x)$; and wherein if $h(x) = y$, the server is configured to perform the requested service for the client;

whereby the server avoids computing $h(x)$ until the server receives the answer with a matching identifier.

22. The apparatus of claim 21, wherein the access mechanism is configured to send a parameter, n, along with the random number y to the client, wherein the parameter n varies the amount of computational work involved in computing the preimage x.

23. The apparatus of claim 22, wherein the parameter n specifies that a subset of n bits of $h(x)$ has to match a corresponding subset of n bits of y.

24. The apparatus of claim 21, wherein computing the preimage, x, takes more computational effort than computing $h(x)$, whereby the client is forced to perform more computational work than the server before the server performs the requested service.

25. The apparatus of claim 21, wherein if $y \neq h(x)$, the server is configured to ignore subsequent communications from the client.

26. The apparatus of claim 21, wherein if $y \neq h(x)$, the server is configured to become slower in responding to subsequent communications from the client, distinguished from other clients, as by its source IP address.

27. The apparatus of claim 26, wherein each time the server determines $y \neq h(x)$, the server is configured to double the service time for the

client, distinguished from other clients, as by its source IP address, so that the server spends progressively less time servicing requests for the client.

28. The apparatus of claim 21, wherein the access mechanism is additionally configured to:

generate the random number y and the identifier id_1;

store the random number y and the identifier id_1 at the server; and

upon receiving the answer from the client, to look up id_1 and the random number y at the server.

29. The apparatus of claim 21, wherein $h(x)$ is a hash function.

30. The apparatus of claim 21, wherein the identifier, id_1, is inferred from data related to the communication.